

05-30-00

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
UTILITY PATENT APPLICATION TRANSMITTAL
UNDER 37 CFR 1.53(b)**

05/25/00
JCS682 U.S.
PTO

PTO
JCS682 U.S.
PTO
05/25/00

Address to: Box Patent Application Assistant Commissioner for Patents Washington, DC 20231	<i>Attorney Docket No.</i>	STL9-2000-0058US1
	<i>Inventor(s)</i>	Neal R. Eisenberg, et al.
	<i>Express Mail Label No.</i>	EL290558612US
	<i>Filing Date</i>	May 25, 2000

Title of Application: METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR STORING, RETRIEVING, & USING REMOTE HOST FILE ATTRIBUTES ON A LOCAL FILE SYSTEM

Transmitted with the patent application are the following:

<u>22</u> Page(s)	Specification, Claims and Abstract
<u>4</u> Page(s)	Formal drawings
<u>3</u> Page(s)	Declaration and Power of Attorney
<u>3</u> Page(s)	Assignment of the Invention to International Business Machines Corporation (including Rec. Cover sheet in duplicate)
<u>2</u> Page(s)	Information Disclosure Statement (IDS/PTO 1449) (copies of citations not included in number of pages)
<u>X</u>	Copies of IDS citations
<u>X</u>	Return Receipt Postcard (MPEP 503).

Fee Calculation:

	Claims		Extra	Rate	Fees
Basic Fee					\$690.00
Total Claims	18	-20 =	0	x \$18.00	
Independent Claims	3	-3 =	0	x \$78.00	
Multiple Dependent Claims				+\$260.00	
TOTAL					\$690.00

- X Please charge my Deposit Account No. 09-0460 in the amount of \$ 690.00. **A duplicate copy of this sheet is attached.**
- X The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit **Account 09-0460**. **A duplicate copy of this sheet is attached.**
- X Any filing fees under 37 CFR 1.16 for the presentation of extra claims.
- X Any patent application processing fees under 37 CFR 1.17.

EXPRESS MAIL CERTIFICATE

I hereby certify that the above paper/fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated below and is addressed to the Assistant Commissioner for Patents, Washington, DC 20231.

Date of Deposit: **May 25, 2000**

Person Mailing paper/fee: Jeanette Berry Durbin

Signature: Jeanette B. Durbin

Respectfully submitted,
Neal R. Eisenberg, et al.

Prentiss W. Johnson
Prentiss W. Johnson,

Registration No. #33,123
Attorney for Applicant(s)
Telephone (408) 463-5673
IBM Corporation
Intellectual Property Law
555 Bailey Avenue (J46/G467)
San Jose, CA 95141-1003

09579866-052500

Express mail Label #: EL290558612US.

S P E C I F I C A T I O N

IBM Docket No. STL9-2000-0058

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that We, Neal Eisenberg of San Jose, California and citizen of the United States, Brent Hawks of Hollister, California and citizen of the United States, Gary Mazo of San Jose, California and citizen of the United States, and Ira Sheftman of San Jose, California and citizen of the United States, have invented new and useful improvements in

METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR STORING, RETRIEVING, & USING REMOTE HOST FILE ATTRIBUTES ON A LOCAL FILE SYSTEM

of which the following is a specification:

1
2
3 **METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR**
4 **STORING, RETRIEVING, & USING REMOTE HOST FILE ATTRIBUTES ON A**
5 **LOCAL FILE SYSTEM**
6
7
8
9
10

11 A portion of the Disclosure of this patent document contains material which is subject
12 to copyright protection. The copyright owner has no objection to the facsimile reproduction by
13 anyone of the patent document or the patent disclosure, as it appears in the Patent and
14 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to computer file systems, and more particularly to retrieving, storing, and using file attributes from one file system in another file system.

2. Description of the Related Art

Data processing systems usually store information in files. A file is a named set or collection of records, logical records, data, or information stored, retrieved, or processed as a unit. Such a file may have various characteristics, and these characteristics may be described by an attribute. An attribute is a specific characteristic that identifies and describes properties of a managed object. The characteristic can be determined, and possibly changed, through operations on the managed object. For example, a file may have attributes that define it as hidden or read-only. Thus, any of the attributes that describe the characteristics of a file are known as file attributes.

Not all file systems share a common identical set of file attributes. For example, a first file system may provide file attributes such as read-only or hidden; whereas, a second file system may not support these file attributes. If a user of the second file system attempts to access files on the first file system, then the second file system will not recognize or appropriately process the file attributes from the first file system.

More specifically, in workstation file systems, externally accessible file attributes are those that can be queried and/or modified by an application programmer using a public application programming interface (PAPI) as opposed to internal functions of the operating

1 system. These are normally implemented as bits that are set on or off, or as integer values in
2 fixed size multi-bit fields such as a four-byte integer. The number of attributes is limited by
3 implementation of the native workstation file system. Installable distributed file systems (that
4 is, file systems that can access files on remote computers) can present difficulties when
5 attempting to convey file attributes that are not present in the local native workstation file
6 system.

7
8 Conventional file attributes in workstation operating systems such as the Windows
9 95™ operating system and the Windows 98™ operating system have been carried over from
10 earlier operating systems such as Disk Operating System (DOS) (Microsoft, Windows,
11 Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United
12 States, other countries, or both.). Modern operating systems such as the OS/2® operating
13 system and the Windows NT™ operating system introduced "extended" attributes that are
14 represented by an extended set of flags or integer values. Remote host file attributes (such as
15 record format, logical record length, or sequence numbers in MVS® data sets) can be used by
16 the workstation tools (such as workstation textual editors, for example the IBM® LPEX
17 editor) to provide additional information to the user and additional functionality to the tools.
18 (IBM®, MVS®, and OS/2® are registered trademarks of International Business Machines
19 Corporation in the United States, other countries, or both.). For example, a textual editor can
20 detect use of sequence numbers in an MVS file and adjust its behavior so that editing of the
21 sequence number area in the file is prohibited by the user. The editor then can automatically
22 adjust sequence numbers when the lines in the file are added or deleted on the workstation.
23 Remote file attributes cannot be represented with workstation file system's conventional or
24 extended file attributes. Existing methods of providing extended information in workstation
25 files normally apply only to files that physically reside on the workstation file systems. Such
26 attributes are normally stored as integral part of the file. Those files are textual files whose
27 formats are specific to the editors that use those formats. For example, the Microsoft™ Word
28 editor uses its own file format, and the Lotus® Word Pro® editor uses yet another format
29 (Lotus® and Word Pro® are registered trademarks of Lotus Development Corporation.).

SUMMARY OF THE INVENTION

The present invention comprises a method, system, article of manufacture, and/or program product for retrieving, storing, and accessing remote file attributes for use on a data processing system's installable file system. The remote file attributes are first obtained on a remote host by a communication program located on the remote host. The remote file attributes are then transferred to a client communication program on the data processing system via a message, preferably a Hyper Text Transfer Protocol (HTTP) message. The client communication program uses the data processing system's shared storage mechanisms to save the remote file attributes and to make them available to other processes executing on the workstation. The remote file attributes have an associated lifetime or duration on the data processing system. This lifetime is pre-determined by the client communication program as to not to exceed a certain time limit, and the remote file attributes are cleared when this maximum lifetime is reached. This clearing causes periodic synchronization between the remote file attributes stored on the data processing system and the remote file attributes stored on the host.

One aspect of a preferred embodiment of the present invention provides an extensible file access method for accessing a foreign file system from a workstation data processing system with a native file system, said foreign file system being located on a remote data processing system, said foreign file system having a set of foreign file attributes corresponding to each of a plurality of files in the foreign file system, and said native file system having a set of native file attributes corresponding to each of a plurality of files in the native file system, said method comprising the steps of: generating a request from a client on the workstation to the remote data processing system to open a foreign file in the foreign file system; opening of the foreign file by the foreign file system; sending of the file attributes of the foreign file, hereinafter foreign file attributes, to the workstation; storing of the foreign file attributes by the workstation; accessing of the foreign file attributes stored in the workstation by the workstation client to process the foreign file; and processing by the workstation client the foreign file using the stored foreign file attributes.

1
2 In accordance with another aspect of a preferred embodiment of the present invention,
3 a subset of the foreign file attributes which are equivalent to a corresponding subset of file
4 attributes of the native file system is determined, the subset of the foreign file attributes
5 hereinafter known as conventional file attributes; the conventional file attributes are returned to
6 the client; and a remaining subset of the foreign file attributes which are not equivalent to a
7 corresponding subset of file attributes of the native file system are stored, the remaining subset
8 of the foreign file attributes hereinafter known as extended file attributes.
9

10 In accordance with another aspect of a preferred embodiment of the present invention,
11 the client accesses the foreign file via a protocol of the native file system, the accessing being
12 performed in a similar manner to accessing a native file system file; and the client accesses the
13 foreign file by use of the extended file attributes, the accessing being performed via a protocol
14 different from the native file system protocol.
15

16 In accordance with another aspect of a preferred embodiment of the present invention,
17 an expiration timer corresponding to the extended file attributes is started; and the extended file
18 attributes are removed from the workstation storage after the expiration of the expiration timer.
19

20 In accordance with another aspect of a preferred embodiment of the present invention,
21 the sending of the foreign file attributes is performed by a web server located on the remote
22 system, the web server being capable of sending and receiving messages via a network.
23

24 In accordance with another aspect of a preferred embodiment of the present invention,
25 the extended file attributes are stored in a shared memory portion of the workstation storage
26 which is accessible by the client and other workstation processes; a unique handle is associated
27 with the extended file attributes; and the unique handle is provided to a workstation process to
28 enable the workstation process to access the extended file attributes.
29

1 A preferred embodiment of the present invention has the advantage of providing
2 improved extended file attributes.

3
4 A preferred embodiment of the present invention has the advantage of providing
5 additional information beyond that provided by native file attributes.

6
7 A preferred embodiment of the present invention has the advantage of allowing
8 additional functionality using the additional information beyond that provided by native file
9 attributes.

10
11 A preferred embodiment of the present invention has the advantage of allowing non-
12 native extended file attributes to be used on a computer system.

13
14 A preferred embodiment of the present invention has the advantage of allowing host
15 extended file attributes to be used on a workstation.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the Description of the Preferred Embodiment in conjunction with the attached Drawings, in which:

Figure 1 is a block diagram of a distributed computer system used in performing the method of the present invention, forming part of the apparatus of the present invention, and which may use the article of manufacture comprising a computer-readable storage medium having a computer program embodied in said medium which may cause the computer system to practice the present invention;

Figure 2 is a block diagram of a portion of memory of the distributed computer system; and

Figures 3 and 4 are flowcharts illustrating the operations preferred in carrying out a preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring first to **Figure 1**, there is depicted a graphical representation of a data processing system **8**, which may be utilized to implement the present invention. As may be seen, data processing system **8** may include a plurality of networks, such as Local Area Networks (LAN) **10** and **32**, each of which preferably includes a plurality of individual computers **12** and **30**, respectively. Of course, those skilled in the art will appreciate that a plurality of Intelligent Work Stations (IWS) coupled to a host processor may be utilized for each such network. Each said network may also consist of a plurality of processors coupled via a communications medium, such as shared memory, shared storage, or an interconnection network. As is common in such data processing systems, each individual computer may be coupled to a storage device **14** and/or a printer/output device **16** and may be provided with a pointing device such as a mouse **17**.

The data processing system **8** may also include multiple mainframe computers, such as mainframe computer **18**, which may be preferably coupled to LAN **10** by means of communications link **22**. The mainframe computer **18** may also be coupled to a storage device **20** which may serve as remote storage for LAN **10**. Similarly, LAN **10** may be coupled via communications link **24** through a sub-system control unit/communications controller **26** and communications link **34** to a gateway server **28**. The gateway server **28** is preferably an IWS which serves to link LAN **32** to LAN **10**.

With respect to LAN **32** and LAN **10**, a plurality of documents or resource objects may be stored within storage device **20** and controlled by mainframe computer **18**, as resource manager or library service for the resource objects thus stored. Of course, those skilled in the art will appreciate that mainframe computer **18** may be located a great geographic distance from LAN **10** and similarly, LAN **10** may be located a substantial distance from LAN **32**. For example, LAN **32** may be located in California while LAN **10** may be located within North Carolina and mainframe computer **18** may be located in New York.

1 Software program code which employs the present invention is typically stored in the
2 memory of a storage device **14** of a stand alone workstation or LAN server from which a
3 developer may access the code for distribution purposes, the software program code may be
4 embodied on any of a variety of known media for use with a data processing system such as a
5 diskette or CD-ROM or may be distributed to users from a memory of one computer system
6 over a network of some type to other computer systems for use by users of such other systems.
7 Such techniques and methods for embodying software code on media and/or distributing
8 software code are well-known and will not be further discussed herein.
9

10 As will be appreciated upon reference to the foregoing, it is often desirable for a user to
11 perform host application development on a workstation **12** in lieu of performing the application
12 development on the host **18** itself. Remote Edit/Compile/Debug provides such a workstation
13 environment for performing the edit, compile, and debug tasks associated with host application
14 development. Host application parts, such as COBOL source code, COBOL copy books, and
15 host JCL, may be stored in PDS or PDSE data sets on storage device **20** connected to the host
16 **18**. The Remote Edit/Compile/Debug workstation environment allows these files to be
17 accessed and used on the workstation **12**. The present invention provides for such access and
18 use of host files on the workstation **12** by retrieving file attributes associated with files stored
19 on the file system **20** of the host **18**, storing these remote file attributes on the workstation **12**,
20 and making the remote file attributes available to processes executing on the workstation **12** to
21 support a scenario such as the Remote Edit/Compile/Debug.
22

23 Referring next to **Figure 2**, a preferred embodiment of the present invention is
24 illustrated. This embodiment comprises a method for storing and accessing remote file
25 attributes **205** on a workstation's **210** installable file system **215**. In the present invention, the
26 remote file attributes **220** are first obtained on the remote host **225** by a communication
27 program **230** in response to a request **235** from the client communication program **240** on the
28 workstation **210**. The remote file attributes are then transferred to the client communication
29 program **240** via a message **245**. The client communication program **240** uses the

1 workstation's **210** shared storage **250** mechanisms to save the attributes **205** and make them
2 available to other processes **260** and **265** on the workstation **210**. The lifetime of the attributes
3 **205** on the workstation **210** is pre-determined by the client communication program **240** as to
4 not to exceed a certain time limit after which the attributes **205** are cleared. This clearing is
5 necessary to force periodic synchronization between corresponding workstation attributes **205**
6 and host attributes **220**.

7
8 Referring now to **Figures 3** and **4**, the flowcharts illustrate the operations preferred in
9 carrying out the preferred embodiment of the present invention. In the flowcharts, the
10 graphical conventions of a diamond for a test or decision and a rectangle for a process or
11 function are used. These conventions are well understood by those skilled in the art, and the
12 flowcharts are sufficient to enable one of ordinary skill to write code in any suitable computer
13 programming language.

14
15 Referring first to **Figure 3**, the process **300** of supporting remote file attributes on the
16 workstation file system begins at process block **310**. Thereafter, process block **320** generates a
17 request **235** from a client communication program **240** on the workstation **210** to the remote
18 data processing system **225** to open a foreign file **270** in the foreign file system **275**. This
19 request **235** may result from an editor attempting to edit the foreign file or remote file.
20 Responsive to the remote host's receiving of the request **235** to open the foreign file **270**,
21 process block **330** causes the foreign file system **275** to open the foreign file **270**. Process
22 block **330** also causes the remote host **225** to obtain the file attributes **220** of the opened foreign
23 file **270**. The file attributes **220** may be of a general type **280** such as creation date,
24 modification date, or size. Alternatively, the file attributes may be of a type **285** specific to
25 certain types of host files such as record format, record length, revision/modification level, and
26 sequence numbers. Thereafter, process block **340** sends **245** the file attributes of the foreign
27 file, hereinafter foreign file attributes, to the workstation **210**. Preferably, the foreign file
28 attributes are sent by a host communication process **230** to the workstation **210** as part of a
29 response **245** to the request **235** to open the file **270**. The host communication process **230** is

preferably a web server capable of sending and receiving HTTP messages. The response is preferably sent in a form of an HTTP message.

The HTTP response **245** is received by the workstation communication program **240**, preferably an IBM Foreign File System client process. Process block **350** determines a subset of the foreign file attributes which are equivalent to a corresponding subset of file attributes of the native file system, the subset of the foreign file attributes hereinafter known as conventional file attributes **255**. Responsive to this determination, process block **360** returns the conventional file attributes **255** to the client operating system as part of a standard file system protocol, preferably, Server Message Block (SMB) protocol. Process block **370** stores a remaining subset of the foreign file attributes which are not equivalent to a corresponding subset of file attributes of the native file system, the remaining subset of the foreign file attributes hereinafter known as extended file attributes **205**. The extended host file attributes **205** are stored in a persistent shared storage **250**, preferably a shared memory segment of the workstation operating system which is accessible by the client **240** and other workstation processes **260** and **265**. Thereafter, processing continues to process block **410** on **Figure 4**. This processing path is illustrated by flowchart connectors **A**, **380** on **Figure 3** and **405** on **Figure 4**.

To enable interested workstation processes **260** and **265** to access the extended file attributes **205** stored in the persistent shared storage **250**, process block **410** associates a unique handle **295** with the extended file attributes **205** and provides the unique handle **295** to the interested workstation processes **260** and **265**. All interested processes **260** and **265** on the workstation **210** obtain a unique name and, through it, a unique handle **295** to the shared memory segment **250** containing the extended file attributes **205** of the file. The unique name of the shared memory segment within the client system is unique because it is derived from the universal naming convention (UNC) name of the file. After the extended file attributes **205** are stored in the persistent shared storage **250**, process block **420** starts an expiration timer corresponding to the extended file attributes. The expiration timer allows the extended file

1 attributes **205** to expire and to become refreshed the next time a workstation process requests to
2 open the host file. The expiration timer also synchronizes the extended attribute information
3 every time the file is opened, and if the expiration timer expires, then the information will be
4 cleared and the shared memory segment will be freed.

5
6 In process block **430**, the workstation client **260** accesses the foreign file attributes **205** by
7 accessing the named persistent shared storage **250** to process the foreign file **290**. In process
8 block **440**, the client then accesses the foreign file **290** via a protocol of the native file system
9 **215**, the accessing being performed in a similar manner to accessing a native file system file.
10 Alternatively, the client **260** may access the foreign file **290** by use of the extended file
11 attributes **205**, the accessing being performed via a protocol different from the native file
12 system protocol, as shown in process block **450**.

13
14 In decision block **460**, the workstation client communication process **240** determines if
15 the extended file attributes expiration timer has expired. If not, process block **470** then waits
16 for the extended file attributes to expire. Referring back to process block **460**, if the expiration
17 timer has expired, then the workstation client communication program **240** in process block
18 **480** clears or removes the extended file attributes **205** from the workstation storage **250** after
19 the expiration of the expiration timer. The process then ends at process block **490**.

20
21 Using the foregoing specification, the invention may be implemented using standard
22 programming and/or engineering techniques using computer programming software, firmware,
23 hardware or any combination or sub-combination thereof. Any such resulting program(s),
24 having computer readable program code means, may be embodied within one or more
25 computer usable media such as fixed (hard) drives, disk, diskettes, optical disks, magnetic tape,
26 semiconductor memories such as read-only memory (ROM), programmable read-only memory
27 (PROM), etc., or any memory or transmitting device, thereby making a computer program
28 product, i.e., an article of manufacture, according to the invention. The article of manufacture
29 containing the computer programming code may be made and/or used by executing the code

1 directly or indirectly from one medium, by copying the code from one medium to another
2 medium, or by transmitting the code over a network. An apparatus for making, using, or
3 selling the invention may be one or more processing systems including, but not limited to,
4 central processing unit (CPU), memory, storage devices, communication links, communication
5 devices, servers, input/output (I/O) devices, or any sub-components or individual parts of one
6 or more processing systems, including software, firmware, hardware or any combination or
7 sub-combination thereof, which embody the invention as set forth in the claims.
8

9 User input may be received from the keyboard, mouse, pen, voice, touch screen, or any
10 other means by which a human can input data to a computer, including through other programs
11 such as application programs.
12

13 One skilled in the art of computer science will easily be able to combine the software
14 created as described with appropriate general purpose or special purpose computer hardware to
15 create a computer system and/or computer sub-components embodying the invention and to
16 create a computer system and/or computer sub-components for carrying out the method of the
17 invention. Although the present invention has been particularly shown and described with
18 reference to a preferred embodiment, it should be apparent that modifications and adaptations
19 to that embodiment may occur to one skilled in the art without departing from the spirit or
20 scope of the present invention as set forth in the following claims.

CLAIMS

We claim:

1. An extensible file access method for accessing a foreign file system from a local data processing system with a native file system, said foreign file system being located on a remote data processing system, said foreign file system having a set of foreign file attributes corresponding to each of a plurality of files in the foreign file system, and said native file system having a set of native file attributes corresponding to each of a plurality of files in the native file system, said method comprising the steps of:

generating a request from a client on the local data processing system to the remote data processing system to open a foreign file in the foreign file system;

opening of the of the foreign file by the foreign file system;

sending of the file attributes of the foreign file, hereinafter foreign file attributes, to the local data processing system;

storing of the foreign file attributes by the local data processing system;

accessing of the foreign file attributes stored in the local data processing system by the local data processing system client to process the foreign file; and

processing by the local data processing system client the foreign file using the stored foreign file attributes.

2. The method of claim 1 further comprising the steps of:

determining a subset of the foreign file attributes which are equivalent to a corresponding subset of file attributes of the native file system, the subset of the foreign file attributes hereinafter known as conventional file attributes;

returning the conventional file attributes to the client; and

storing a remaining subset of the foreign file attributes which are not equivalent to a corresponding subset of file attributes of the native file system, the remaining subset of the foreign file attributes hereinafter known as extended file attributes.

3. The method of claim 2 further comprising the steps of:
accessing of the foreign file by the client via a protocol of the native file system, the
accessing being performed in a similar manner to accessing a native file system file; and
accessing of the foreign file by the client by use of the extended file attributes, the
accessing being performed via a protocol different from the native file system protocol.

4. The method of claim 2 wherein the storing step further comprises:
starting an expiration timer corresponding to the extended file attributes; and
removing the extended file attributes from the local data processing system storage after
the expiration of the expiration timer.

5. The method of claim 2 wherein the sending of the foreign file attributes is performed by
a web server located on the remote system, the web server being capable of sending and
receiving messages via a network.

6. The method of claim 2 further comprising the steps of:
storing the extended file attributes in a shared memory portion of the local data
processing system storage which is accessible by the client and other local data processing
system processes;
associating a unique handle with the extended file attributes; and
providing the unique handle to a local data processing system process to enable the
local data processing system process to access the extended file attributes.

1 7. An article of manufacture for use in a computer system for accessing a foreign file
2 system from a local data processing system data processing system with a native file system,
3 said foreign file system being located on a remote data processing system, said foreign file
4 system having a set of foreign file attributes corresponding to each of a plurality of files in the
5 foreign file system, and said native file system having a set of native file attributes
6 corresponding to each of a plurality of files in the native file system, said article of manufacture
7 comprising a computer-readable storage medium having a computer program embodied in said
8 medium which causes the computer system to execute the method steps comprising:

9 generating a request from a client on the local data processing system to the remote data
10 processing system to open a foreign file in the foreign file system;

11 opening of the of the foreign file by the foreign file system;

12 sending of the file attributes of the foreign file, hereinafter foreign file attributes, to the
13 local data processing system;

14 storing of the foreign file attributes by the local data processing system;

15 accessing of the foreign file attributes stored in the local data processing system by the
16 local data processing system client to process the foreign file; and

17 processing by the local data processing system client the foreign file using the stored
18 foreign file attributes.

19 8. The article of manufacture of claim 7 wherein the method steps further comprise the
2 steps of:

3 determining a subset of the foreign file attributes which are equivalent to a
4 corresponding subset of file attributes of the native file system, the subset of the foreign file
5 attributes hereinafter known as conventional file attributes;

6 returning the conventional file attributes to the client; and

7 storing a remaining subset of the foreign file attributes which are not equivalent to a
8 corresponding subset of file attributes of the native file system, the remaining subset of the
9 foreign file attributes hereinafter known as extended file attributes.

9. The article of manufacture of claim 8 wherein the method steps further comprise the steps of:

accessing of the foreign file by the client via a protocol of the native file system, the accessing being performed in a similar manner to accessing a native file system file; and
accessing of the foreign file by the client by use of the extended file attributes, the accessing being performed via a protocol different from the native file system protocol.

10. The article of manufacture of claim 8 wherein the storing step further comprises:
starting an expiration timer corresponding to the extended file attributes; and
removing the extended file attributes from the local data processing system storage after the expiration of the expiration timer.

11. The article of manufacture of claim 8 wherein the sending of the foreign file attributes is performed by a web server located on the remote system, the web server being capable of sending and receiving messages via a network.

12. The article of manufacture of claim 8 wherein the method steps further comprise the steps of:

storing the extended file attributes in a shared memory portion of the local data processing system storage which is accessible by the client and other local data processing system processes;

associating a unique handle with the extended file attributes; and

providing the unique handle to a local data processing system process to enable the local data processing system process to access the extended file attributes.

1 13. A distributed computer system for accessing a foreign file system from a local data
2 processing system with a native file system, said foreign file system being located on a remote
3 data processing system, said foreign file system having a set of foreign file attributes
4 corresponding to each of a plurality of files in the foreign file system, and said native file
5 system having a set of native file attributes corresponding to each of a plurality of files in the
6 native file system, said distributed computer system comprising:

7 a requestor for generating a request from a client on the local data processing system to
8 the remote data processing system to open a foreign file in the foreign file system;

9 a foreign file which can be opened by the foreign file system;

10 a sender for sending the file attributes of the foreign file, hereinafter foreign file
11 attributes, to the local data processing system;

12 storage for storing of the foreign file attributes by the local data processing system;

13 a file access for accessing the foreign file attributes stored in the local data processing
14 system by the local data processing system client to process the foreign file; and

15 a processor for processing by the local data processing system client the foreign file
16 using the stored foreign file attributes.

17 14. The distributed computer system of claim 13 further comprising:

18 a comparator for determining a subset of the foreign file attributes which are equivalent
19 to a corresponding subset of file attributes of the native file system, the subset of the foreign
20 file attributes hereinafter known as conventional file attributes;

21 a data transfer for returning the conventional file attributes to the client; and

22 storage for storing a remaining subset of the foreign file attributes which are not
23 equivalent to a corresponding subset of file attributes of the native file system, the remaining
24 subset of the foreign file attributes hereinafter known as extended file attributes.

1 15. The distributed computer system of claim 14 further comprising:
2 a file access for accessing by the client the foreign file via a protocol of the native file
3 system, the accessing being performed in a similar manner to accessing a native file system
4 file; and
5 a file access for accessing by the client the foreign file by use of the extended file
6 attributes, the accessing being performed via a protocol different from the native file system
7 protocol.

1 16. The distributed computer system of claim 14 wherein the storage further comprises:
2 an expiration timer corresponding to the extended file attributes; and
3 storage access for removing the extended file attributes from the local data processing
4 system storage after the expiration of the expiration timer.

1 17. The distributed computer system of claim 14 wherein the sender of the foreign file
2 attributes is a web server located on the remote system, the web server being capable of
3 sending and receiving messages via a network.

1 18. The distributed computer system of claim 14 further comprising:
2 a shared memory portion of the local data processing system storage which is accessible
3 by the client and other local data processing system processes for storing the extended file
4 attributes;
5 a unique handle associated with the extended file attributes; and
6 a local data processing system process which uses the unique handle to enable the local
7 data processing system process to access the extended file attributes.

ABSTRACT

**METHOD OF, SYSTEM FOR, AND COMPUTER PROGRAM PRODUCT FOR
STORING, RETRIEVING, & USING REMOTE HOST FILE ATTRIBUTES ON A
LOCAL FILE SYSTEM**

Remote file attributes are retrieved, stored, and accessed for use on a local data processing system's installable file system. The remote file attributes are first obtained on a remote host by a communication program located on the remote host. The remote file attributes are then transferred to a client communication program on the local data processing system via a message. The client communication program uses the local data processing system's shared storage mechanisms to save the remote file attributes and to make them available to other processes executing on the local data processing system. The remote file attributes have an associated lifetime or duration on the local data processing system. This lifetime is pre-determined by the client communication program as to not to exceed a certain time limit, and the remote file attributes are cleared when this maximum lifetime is reached. This clearing causes periodic synchronization between the remote file attributes stored on the local data processing system and the remote file attributes on the host.

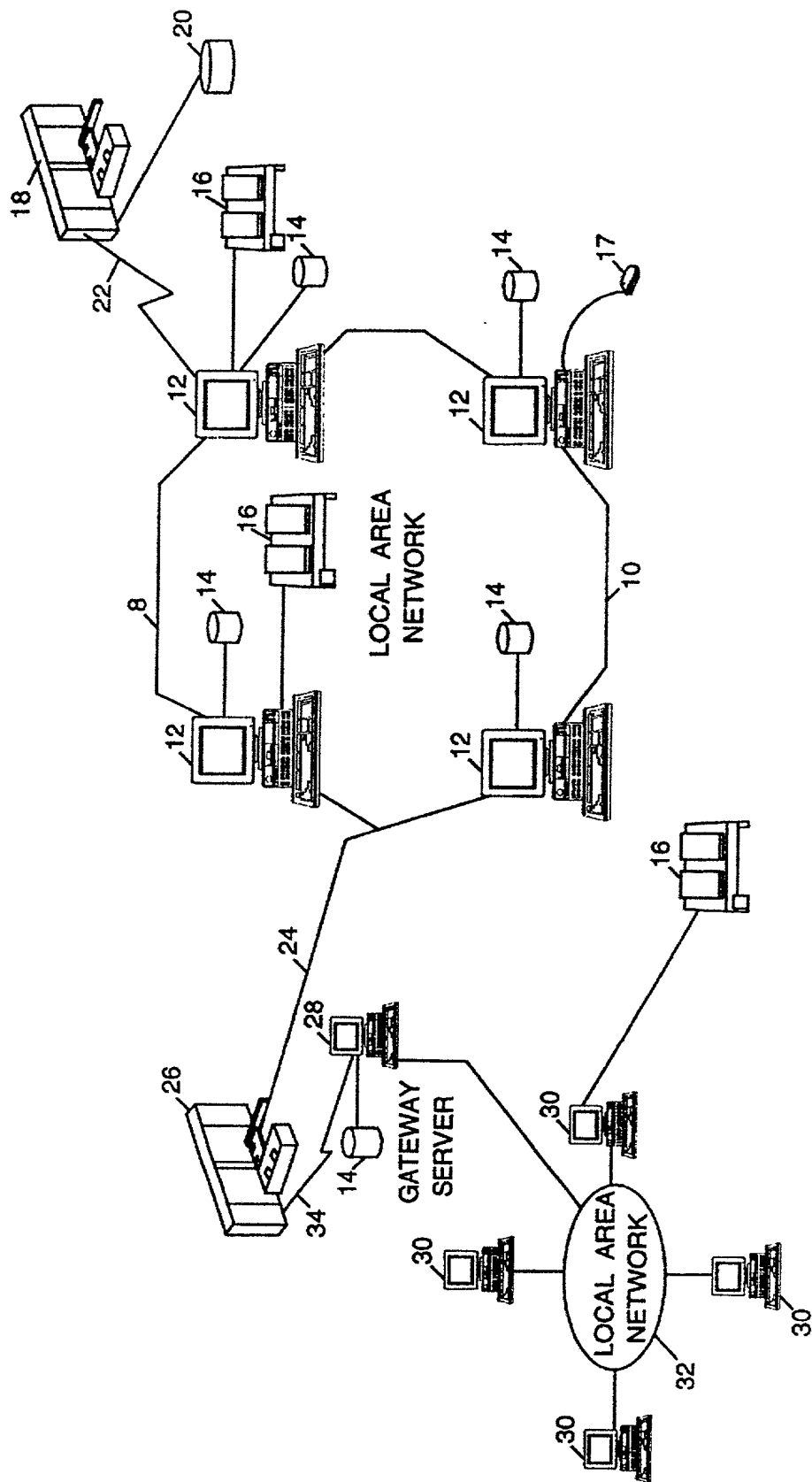


FIG. 1

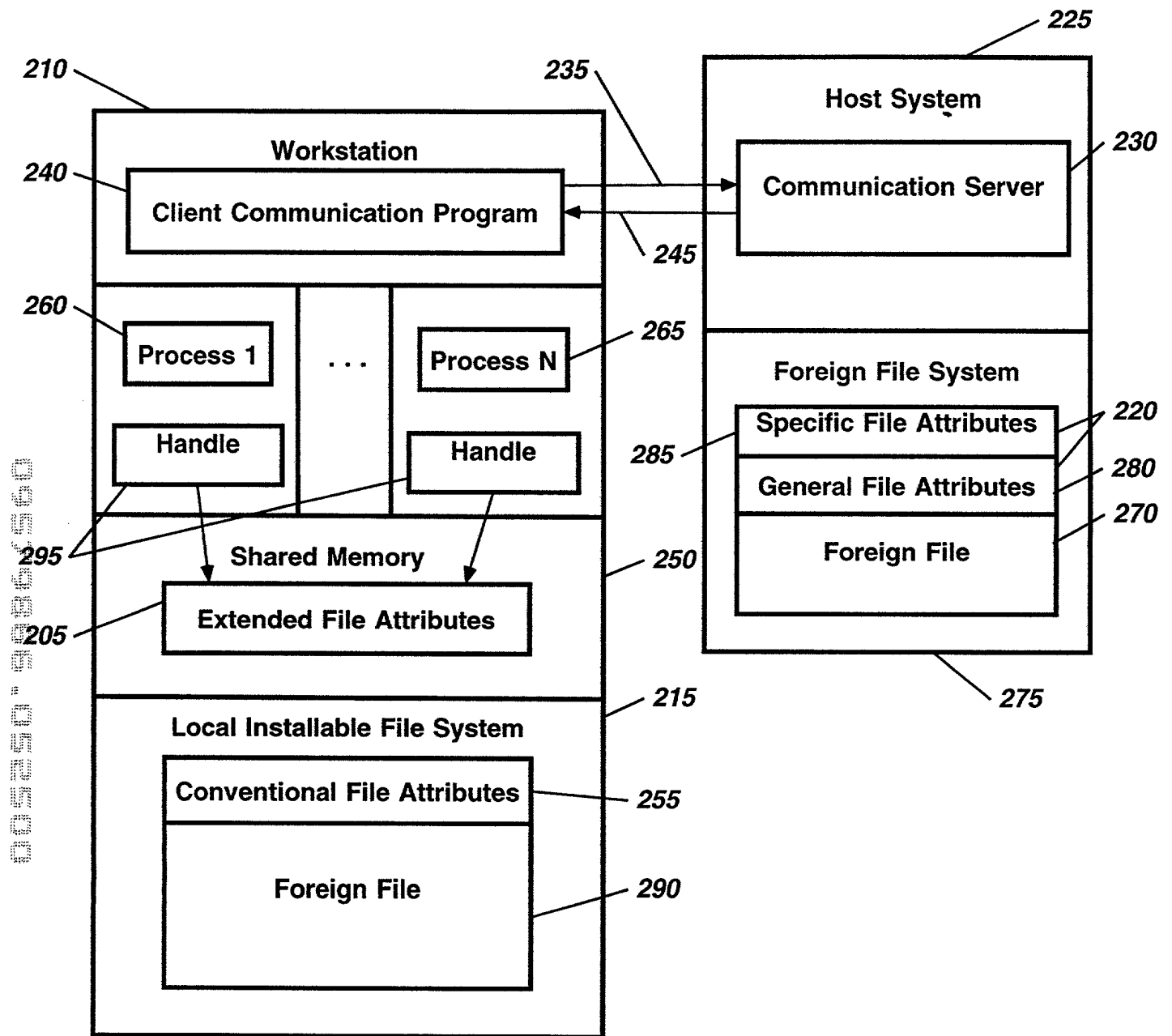


Fig. 2

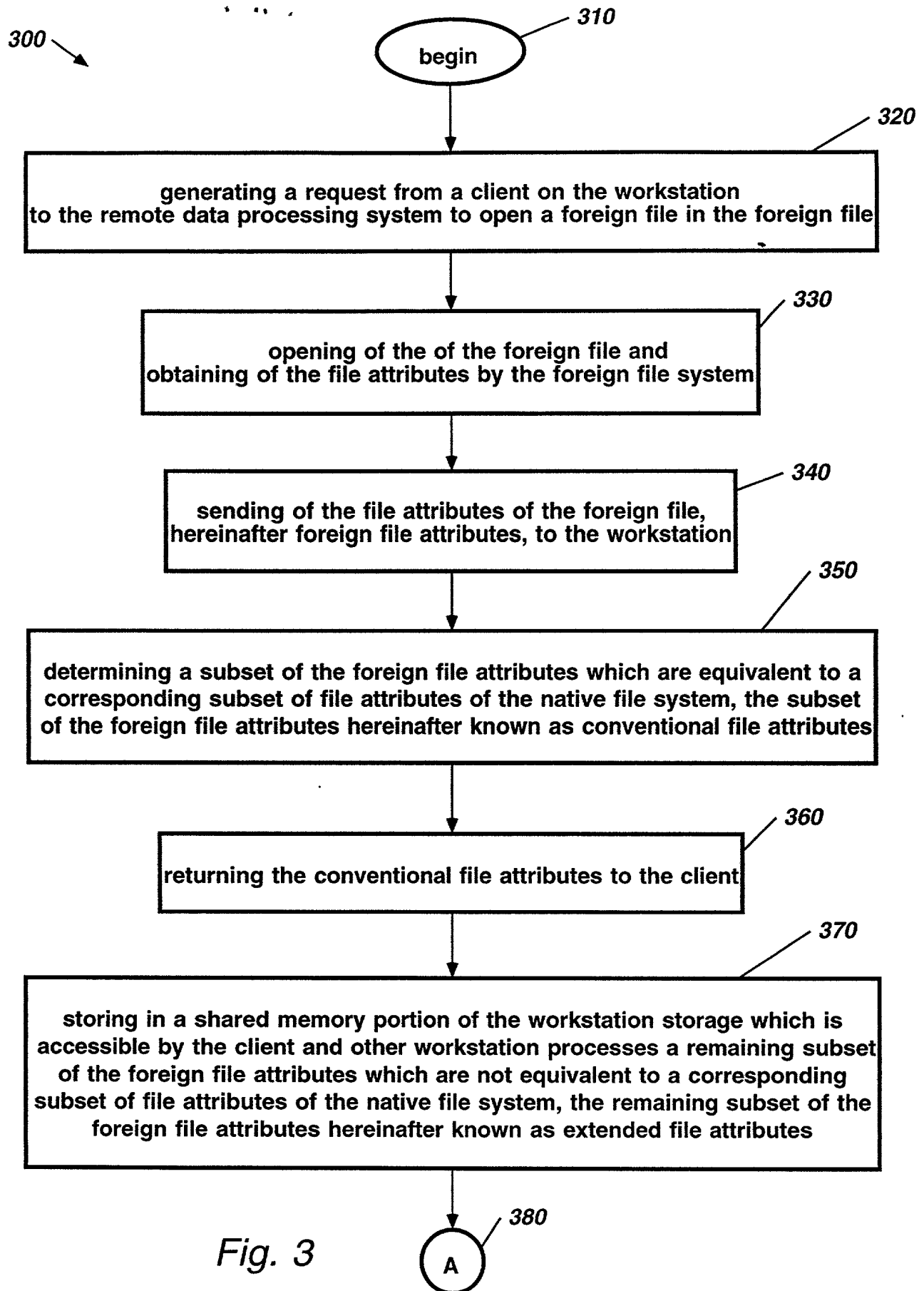


Fig. 3

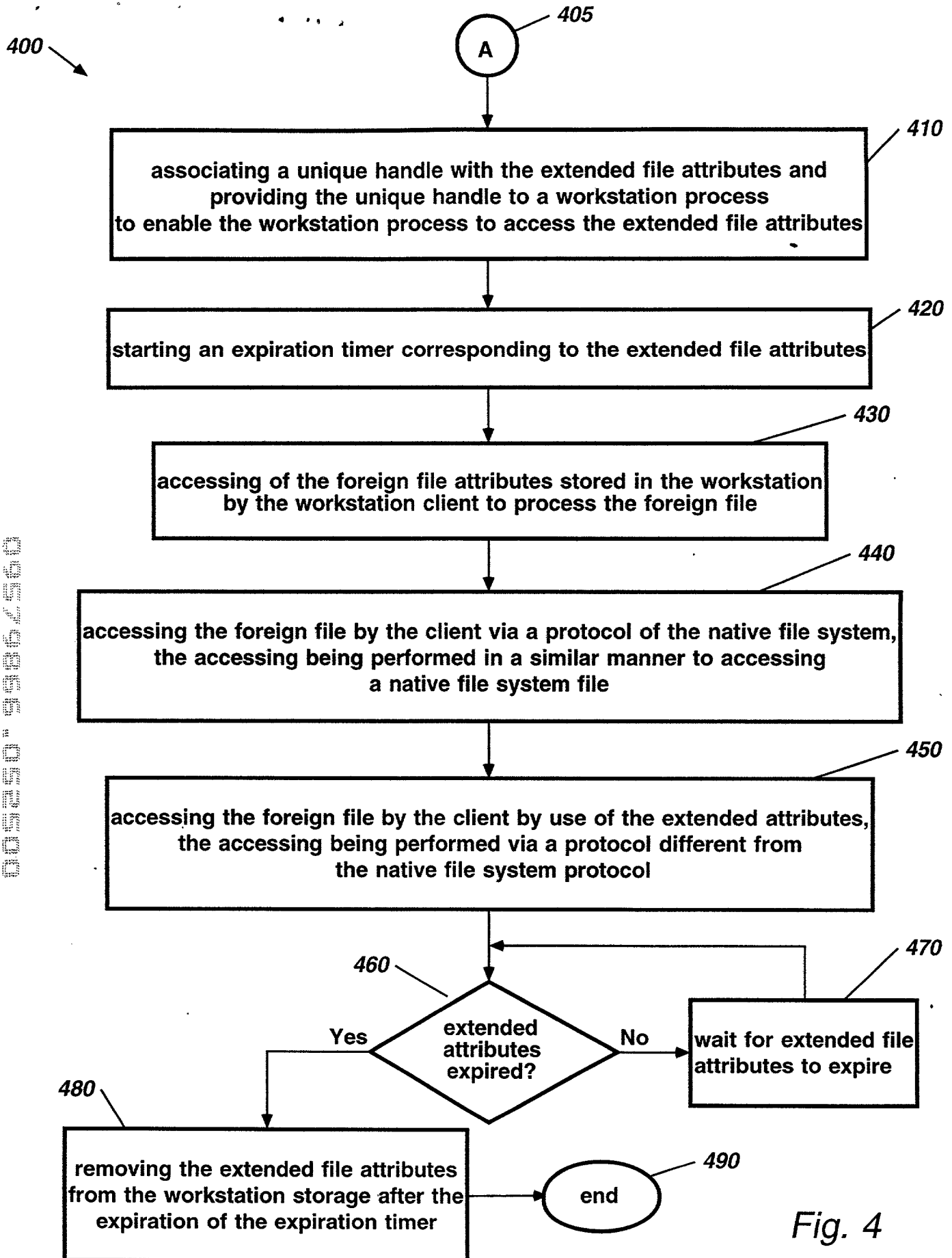


Fig. 4

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0058US1

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Method of, System for, and Computer Program Product for Storing, Retrieving, & Using Remote Host File Attributes on a Local File System

the specification of which (check one)

☒ is attached hereto.

☐ was filed on _____

as Application Serial No. _____

and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

Priority Claimed

#	None			Yes	No
	(Number)	(Country)	(Day/Month/Year Filed)		

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

None		
(Application Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0058US1

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (list name and registration number)

Romualdas Strimaitis (#35,697)	Christopher A. Hughes (#26,914)
Timothy M. Farrell (#37,321)	Edward A. Pennington (#32,588)
Ingrid M. Foerster (#36,511)	John E. Hoel (#26,279)
Prentiss W. Johnson (# 33,123)	Joseph C. Redmond, Jr. (#18,753)

Send correspondence to:

Prentiss Wayne Johnson
Attorney
IBM Corporation
Intellectual Property Law
555 Bailey Avenue (J46/G467)
San Jose CA 95141-1003

Direct Telephone Calls to: (name and telephone number) Prentiss W. Johnson, 408-463-5673

Full name of sole or first joint-inventor: Neal R. Eisenberg

Inventor's signature:

Neal R. Eisenberg

Date:

May 25, 2000

Residence: 787 Colleen Drive, San Jose, California 95123

Citizenship: USA

Post Office Address: Same

Full name of second joint-inventor: Brent C. Hawks

Inventor's signature:

Brent C. Hawks

Date:

25 May 2000

Residence: 310 Howard Court, Hollister, California 95023

Citizenship: USA

Post Office Address: Same

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

DOCKET: STL9-2000-0058US1

Full name of third joint-inventor: Gary I. Mazo

Inventor's signature:

Gary I. Mazo

Date:

May 25, 2000

Residence: 2861 Ruby Terrace, San Jose, California 95148

Citizenship: USA

Post Office Address: Same

Full name of fourth joint-inventor: Ira L. Sheftman

Inventor's signature:

Ira L. Sheftman

Date:

5/25/2000

Residence: 7442 Tulare Hill Drive, San Jose, California 95139

Citizenship: USA

Post Office Address: Same